An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 41 (2001) 1343–1363
www.elsevier.nl/locate/camwa

Solution of Nonlinear Problems in Applied Sciences by Generalized Collocation Methods and Mathematica

N. Bellomo and E. De Angelis
Department of Mathematics
Politecnico Torino
Torino 10129, Italy

L. Graziano and A. Romano
Department of Mathematics
University of Napoli
Napoli 80126, Italy

(Received and accepted July 2000)

Abstract—This paper deals with the developments of mathematical methods for the discretization of continuous models and the solution of nonlinear problems of interest in applied sciences. The contents refer to developments of the differential quadrature method which leads to the so-called generalized collocation methods. The method is developed and applied to the solution of initial-boundary value problems. The computational problems are technically solved with *Mathematica*.

Keywords—Nonlinear problems, Nonlinear sciences, Evolution equations, Sinc functions, Collocation, Interpolation, Spectral methods.

1. INTRODUCTION

Modelling and analysis of physical phenomena in applied sciences often generates nonlinear mathematical problems. Nonlinearity may be an inner feature of the model, i.e., evolution equations with nonlinear terms, or of the problem, i.e., nonlinear boundary conditions. The interplay between applied sciences and mathematics then leads to the development of initial and/or boundary value problems for nonlinear partial differential or integro-differential equations modelling real physical systems.

A well-known and commonly applied solution technique of nonlinear initial-boundary value problems for partial differential equations is the collocation-interpolation method originally proposed as differential quadrature method by Bellman and Casti [1], see also [2]. This method was developed by several authors in the deterministic and stochastic framework, as it was documented in some review papers, e.g., [3,4], devoted to the above topic. In particular, the method can provide a useful discretization of continuous models and efficiently deals with nonlinearities including the ones related to implicit boundary conditions.

Partially supported by GNFM of the Istituto di Alta Matematica.

0898-1221/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved. Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(01)00101-8

The survey [4] provided a detailed report on the application of the original differential quadrature method to several interesting engineering problems. Additional references on applications can be recovered in various papers dedicated to this topic, e.g., [5–10]. This method discretizes the original continuous model (and problem) into a discrete (in space) model, with a finite number of degrees of freedom, while the initial-boundary value problem is transformed into an initial-value problem for ordinary differential equations.

On the other hand, it is known that the method does not generally work in some circumstances. For instance, referring to the Dirichlet problem, the classical Lagrange interpolation may not be useful to deal with problems in unbounded domains and with solutions that are oscillating with high frequency in the space variables. This problem was overcome by a suitable use of Sinc functions, see [11–13], which are characterized by spectral approximation properties.

Sinc methods for nonlinear differential problems were first introduced in [14] and subsequently developed in [15,16]. The analysis and experiments developed in the above cited papers lead to the conclusion that Sinc methods are useful and efficient methods of nonlinear differential problems arising in technology. This paper, which has to be regarded as a complement, for mathematical aspects, to [4], aims to provide an updating of the state of the art and show the technical application of the method to nonlinear problems also with the support of scientific computational softwares. Although this paper is mainly devoted to deal with nonlinearities both in the model and in the problem in one space dimension, application of the methods in two space dimensions will also be described.

The contents are organized in seven sections.

- Section 1 deals with a general introduction to the contents and to the aims of the paper.
- Section 2 describes a one-dimensional mathematical model of hydrodynamic traffic flow. The model is stated in terms of a nonlinear partial differential equation. Some mathematical problems are proposed in order to define some specific examples for the application of the mathematical method.
- Section 3 deals with the solution of initial boundary value problems for nonlinear partial differential equations in one space dimension. The solution method needs, as we shall see, interpolation of functions by Sinc collocation interpolation methods, and transformation of the initial boundary value problem into an initial value problem for ordinary differential equations. The application of the method is practically shown for the problems stated in Section 2.
- Section 4 deals with the treatment of nonlinear boundary conditions. Again, the problems stated in Section 2 become the test for the development of the mathematical method.
- Section 5 provides some notes on the treatment of the mathematical methods by use of the software *Mathematica*. The result of some simulations are practically shown.
- Section 6 reports about mathematical aspects related to Sinc interpolation. It is a collection of *a priori* estimates which may be useful for the computation of the error bounds. Indeed, this section also deals with the estimate of errors and correction terms.
- The last section deals with the analysis of further generalizations and developments of the method. In particular, we deal with solution of problems in two space dimensions, and with alternative interpolation and approximation methods.

In proposing the contents of this paper, it needs to be stated that in alternative to the above method, nonlinear problems can be dealt with by classical methods of applied mathematics, e.g., finite differences [17,18], finite elements methods [19,20], particle methods [21], Galerkin methods [22], Trefftz [23], and so on. In some cases, the above methods can be relatively more efficient than the collocation methods. In any case, the selection of one method with respect to others can be done only for specific models and problems and is not object of speculations in this present paper. As a matter of fact, the selection of the proper method is one of the difficult tasks of applied mathematics and depends mainly on the structure of the problem to be solved.

Similarly, the choice of the software *Mathematica* is also related to the personal experience of the authors [24]. Alternative softwares can be used by the reader according to one's own experience.

2. NONLINEAR TRAFFIC FLOW HYDRODYNAMIC MODELS AND PROBLEMS

This section describes a sample model and the related initial-boundary value problem to be used, at tutorial level, for the applications developed in what follows. This example will be used to relate the application of the Sinc method to nonlinear problems.

The model refers to hydrodynamic traffic flow. The system is of great interest for engineering sciences due to its impact to regional planning and environmental sciences. General references can be recovered in the book by Leutzback [25] and in the review papers [26,27]. The application of the method generates interesting problems with linear and nonlinear boundary conditions, which are a useful test for the application of the mathematical method. In particular, it will be shown how to write the model in a dimensionless form by suitable rescaled variables, considering that the application of the method takes advantages in operating on problems with variables defined over the fundamental interval $[0, 1]$.

Bearing this in mind, consider the one-dimensional flow of vehicles along a road with length ℓ and let

$$n = n(\tau, z) : [0, T] \times [0, \ell] \to \mathbb{R}^+, \tag{2.1}$$

the number density (number of vehicles per unit tract) at time τ, in the point z. A scalar traffic flow model is an evolution equation, consistent with hydrodynamic mass conservation, defining the time and space evolution of n. Considering that mass conservation involves both n and the mass velocity V of vehicles, a self-consistent model can be obtained if a constitutive relation (or better a phenomenologic model) can be proposed to link V to n.

Before dealing with modelling aspects, it is convenient introducing suitable independent and dependent variables rescaled in order to take values in the interval $[0, 1]$. In details, let

$u = n/n_M$	be the dimensionless number density referred to the maximum density n_M corresponding to bump-to-bump vehicles,
$x = z/\ell$	be the dimensionless space variable referred to the length of the road,
$t = \tau/T$	be the dimensionless time variable referred to a suitable observation time T, which will be defined later, and
$v = V/V_M$	be the dimensionless velocity referred to the maximum admissible velocity V_M.

The mathematical model is derived, following [28], according to the following assumptions.

HYPOTHESIS 2.1. *The flow is continuous in one space dimension and the number density* $n = n(\tau, z)$ *is a continuous variable defined* $\forall\, \tau,\ z \in [0, T] \times [0, \ell]$.

HYPOTHESIS 2.2. *The driver adapts instantaneously the velocity of the vehicle to an equilibrium velocity which depends on local fictitious density* u^*, *depending on the real density and density gradient as follows:*

$$u^* = u + \eta(1-u)\,\frac{\partial u}{\partial x}, \tag{2.2}$$

$$v = 1 - u^*, \tag{2.3}$$

where η *is a positive, small, with respect to unity, constant parameter to be identified by phenomenologic observations.*

Mass conservation equation writes

$$\frac{\partial n}{\partial \tau} + \frac{\partial}{\partial z}\,(Vn) = 0, \tag{2.4}$$

while introducing the dimensionless independent and dependent variables yields

$$\frac{\partial u}{\partial t} + \left(\frac{TV_M}{\ell}\right) \frac{\partial}{\partial x}(vu) = 0. \tag{2.5}$$

This equation also indicates that a proper choice of the reference time is the following:

$$\frac{TV_M}{\ell} = 1 \quad \Rightarrow \quad T = \frac{\ell}{V_M}. \tag{2.6}$$

The dimensionless mass conservation equation writes

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(vu) = 0, \tag{2.7}$$

and using the expression of v given by Hypothesis 2.2 yields the hydrodynamic equation

$$\frac{\partial u}{\partial t} = (2u-1)\frac{\partial u}{\partial x} + \eta u(1-u)\frac{\partial^2 u}{\partial x^2} + \eta(1-2u)\left(\frac{\partial u}{\partial x}\right)^2. \tag{2.8}$$

The statement of problems is related to the quantities which can be effectively measured on the boundaries $x = 0$ and $x = 1$. If it is possible to measure the densities, one has a Dirichlet type problem with linear boundary conditions. On the other hand, if it is possible to measure the gradients, one has the Neuman problem. Nonlinear boundary conditions are generated when the measured quantity is the dimensionless flow $q(t,\cdot) = u(t,\cdot)v(t,\cdot)$. The statement of the above problems is the following.

PROBLEM 2.1. DIRICHLET PROBLEM. The initial-boundary value problem for equation (2.8) is stated with initial condition

$$u(0,x) = \varphi(x), \qquad \forall x \in [0,1], \tag{2.9}$$

and Dirichlet boundary conditions

$$u(t,0) = \alpha(t) \qquad \text{and} \qquad u(t,1) = \beta(t), \qquad \forall t \in [0,1], \tag{2.10}$$

where φ is a given function of space, while α and β are given smooth functions of time.

PROBLEM 2.2. NEUMANN PROBLEM. The initial-boundary value problem for equation (2.8) is stated with initial condition (2.9), and boundary conditions

$$\frac{\partial u}{\partial x}(t,0) = \gamma(t) \qquad \text{and} \qquad \frac{\partial u}{\partial x}(t,1) = \delta(t), \qquad \forall t \in [0,1], \tag{2.11}$$

where φ is a given function of space, while γ and δ are given smooth functions of time.

Consider now the problem with conditions at the boundary identified by the flow q. In this case, the relation between u and v is defined by Hypothesis 2.2. Hence,

$$q = u(1-u)\left[1 - \eta\frac{\partial u}{\partial x}\right]. \tag{2.12}$$

It follows an initial-boundary value problem with nonlinear boundary conditions involving both Dirichlet and Neumann conditions. The statement is as follows.

PROBLEM 2.3. NONLINEAR BOUNDARY CONDITIONS. The initial-boundary value problem for equation (2.8) is stated with initial condition (2.9) and boundary conditions defined by the system

$$\begin{aligned} \left[\alpha(t) - \alpha^2(t)\right][1 - \eta\gamma(t)] &= q(t,0), \\ \left[\beta(t) - \beta^2(t)\right][1 - \eta\delta(t)] &= q(t,1). \end{aligned} \tag{2.13}$$

The analysis developed in what follows will technically refer to the above model and problems. We are not interested in improving the above model or in discussing its validity. The interested reader is referred to [28] for such a purpose. The aim is to use the above problems for the practical, and tutorial, application of the mathematical methods proposed in the sections which follow.

3. DIRICHLET AND NEUMANN PROBLEMS IN ONE SPACE DIMENSION

The solution methods described in what follows will be referred to the model described in Section 2 and related to the classical Dirichlet and Neumann problems. Generalizations to nonlinear boundary conditions are dealt with later. Detailed description is given for problems in one space dimension. Problems in more than one space dimension are dealt with as generalization of the above relatively simpler case.

In general, we assume that problems are such that both independent and dependent variables are defined over a bounded domain and are subject to a nondimensionalization procedure, with reference to the minimum and maximum values of each variable so that each one is defined over the interval $[0,1]$. For instance, if the space variables are defined over a rectangle $[x_m, x_M] \times [y_m, y_M]$, then the dependent variable $u = u(t,x,y)$ defines, after a suitable normalization of the dependent variables an application from $[0,1] \times [0,1] \times [0,1]$ into $[0,1]$. When problems are such that the dependent variables are unbounded, then further technicalities will be specifically indicated. Moreover, it is assumed that the solution exists unique and smooth in a suitable function space. Of course, this strong statement has to be properly verified for specific models and problems. In this section, we deal with relatively simpler problems in one space dimension.

We summarize, in this section, the application of the Sinc differential quadrature method to the solution of initial-boundary value problems described by the nonlinear partial differential equation in one space dimension. We first deal with the Dirichlet problem, then generalizations to Neumann and mixed problems will be given. Direct reference is made to the problems described in Section 2.

In details, the application of the method for the solution of Problem 2.1 goes through the following steps.

Step 1. The space variable is discretized into a suitable collocation I_x

$$i = 1,\dots,n : I_x = \{x_1 = 0,\dots,x_i,\dots,x_n = 1\}, \quad x_i = (i-1)h, \qquad h = \frac{1}{n-1}, \tag{3.1}$$

and the dependent variable $u = u(t,x)$ is interpolated and approximated by the values $u_i(t) = u(t,x_i)$ by means of Sinc functions as follows:

$$u(t,x) \cong u^n(t,x) = \sum_{j=1}^{n} S_j(x;h) u_j(t), \tag{3.2}$$

where $u_i(t) = u(t,x_i)$ and

$$S_j(x;h) = \frac{\sin\left((\pi/h)(x - jh)\right)}{(\pi/h)(x - jh)}. \tag{3.3}$$

Step 2. The interpolation defined in Step 1 can be used to approximate the partial derivatives of the variable u in the nodal points of the discretization

$$\frac{\partial^r u}{\partial x^r}(t;x_i) \cong \sum_{j=1}^{n} \frac{d^r S_j}{dx^r}(x_i) u_j(t) = \sum_{j=1}^{n} a_{ji}^{(r)} u_j(t), \tag{3.4}$$

where $r = 1,2,\dots$, and the value of the coefficients depends on the number of collocation points and on the type of collocation. Technical calculations provide the following result for the first two derivative matrices

$$a_{ji}^{(1)} = \frac{(-1)^{i-j}}{h(i-j)}, \qquad a_{ii}^{(1)} = 0, \tag{3.5}$$

$$a_{ji}^{(2)} = 2\,\frac{(-1)^{i-j+1}}{h^2(i-j)^2}, \qquad a_{ii}^{(2)} = -\frac{1}{3}\left(\frac{\pi}{h}\right)^2. \tag{3.6}$$

Step 3. The interpolation and approximation of the space derivatives are replaced into the evolution equation, then particularizing the equation in each node and enforcing the boundary conditions in the first and the last node transforms the initial-boundary value problem into an initial value problem for ordinary differential equations describing the time-evolution of the values $u_i(t)$ of u in the nodes.

The solution of Problem 2.1 is then obtained by solving the following system of ordinary differential equations:

$$\frac{du_i}{dt} = f_i(t, u_1, \ldots, u_n), \qquad i = 1, \ldots, n, \tag{3.7}$$

where

$$f_1 = \alpha_t(t), \tag{3.8}$$

$$\begin{aligned} f_i = (2u_i - 1) &\left[\sum_{j=1, j\neq i}^{n} \frac{(-1)^{i-j}}{h(i-j)} u_j \right] \\ + \eta u_i(1 - u_i) &\left[2 \sum_{j=1, j\neq i}^{n} \frac{(-1)^{i-j+1}}{h^2(i-j)^2} u_j - \frac{1}{3} \left(\frac{\pi}{h}\right)^2 u_i \right] \\ + \eta(1 - 2u_i) &\left[\sum_{j=1, j\neq i}^{n} \frac{(-1)^{i-j}}{h(i-j)} u_j \right]^2, \end{aligned} \tag{3.9}$$

for $i = 2, \ldots, n-1$, and

$$f_n = \beta_t(t), \tag{3.10}$$

where the subscript denotes partial derivatives. We recall that in (3.9) $u_1 = \alpha(t)$ and $u_n = \beta(t)$.

This system has to be linked to the initial conditions

$$u_i(0) = \varphi_i = \varphi(x_i), \tag{3.11}$$

for $i = 1, \ldots, n$, where φ_i has to satisfies the following compatibility conditions with the data $\alpha(t), \beta(t)$:

$$\alpha(0) = \varphi_1, \qquad \beta(0) = \varphi_n. \tag{3.12}$$

The solution of the initial-boundary value problem is then obtained solving the initial value problem (3.7)–(3.10), by suitable methods for ordinary differential equations, see [17, Chapter 2], and interpolating the solution by the method used in Step 2.

The solution of Problem 2.2 can be developed analogously. In this case, the corresponding system of ordinary differential equations has to be linked to the boundary conditions that can be recovered by the solution of the following system:

$$\begin{aligned} \frac{(-1)^{1-n}}{(1-n)} \beta(t) + \sum_{j=2}^{n-1} \frac{(-1)^{1-j}}{(1-j)} u_j(t) &= h\gamma(t), \\ \frac{(-1)^{n-1}}{(n-1)} \alpha(t) + \sum_{j=2}^{n-1} \frac{(-1)^{n-j}}{(n-j)} u_j(t) &= h\delta(t). \end{aligned} \tag{3.13}$$

It can be noted that Sinc interpolations are such that $a_{ii} = 0$. Therefore, the local coordinate does not contribute to space derivative so that the first equation contains only β, while the second one gives α. It may be convenient, for practical calculations deriving, with respect to time, for smooth behaviour of the boundary conditions, both equations (3.13) in order to obtain the solution of Problem 2.2 as the solution of the following system of ordinary differential equations:

$$\frac{du_i}{dt} = f_i(t, u_1, \ldots, u_n), \qquad i = 1, \ldots, n, \tag{3.14}$$

where

$$f_1 = \frac{(n-1)}{(-1)^{n-1}}\left[h\,\delta_t(t) - \sum_{j=2}^{n-1}\frac{(-1)^{n-j}}{(n-j)}f_j(t,u_1,\dots,u_n)\right], \tag{3.15}$$

$$\begin{aligned} f_i = (2u_i - 1)&\left[\sum_{j=1,j\neq i}^{n}\frac{(-1)^{i-j}}{h(i-j)}u_j\right] \\ &+ \eta u_i(1-u_i)\left[2\sum_{j=1,j\neq i}^{n}\frac{(-1)^{i-j+1}}{h^2(i-j)^2}u_j - \frac{1}{3}\left(\frac{\pi}{h}\right)^2 u_i\right] \\ &+ \eta(1-2u_i)\left[\sum_{j=1,j\neq i}^{n}\frac{(-1)^{i-j}}{h(i-j)}u_j\right]^2, \end{aligned} \tag{3.16}$$

for $i = 2,\dots,n-1$, and

$$f_n = \frac{(1-n)}{(-1)^{1-n}}\left[h\,\gamma_t(t) - \sum_{j=2}^{n-1}\frac{(-1)^{1-j}}{(1-j)}f_j(t,u_1,\dots,u_n)\right]. \tag{3.17}$$

REMARK 3.1. The interpolation is exactly satisfied in the nodal points while partial derivatives are only approximated. This induce computational errors which have to be controlled as will be discussed later.

REMARK 3.2. The solution method can be generalized to the analysis of *problems in unbounded domains* with decaying data at infinity. For instance to the initial-boundary value problem for equation (2.8) in the half-space $x \in [0,\infty)$ with initial conditions $\varphi(x)$, and boundary conditions of the type

$$u(0,x) = \alpha(t), \qquad \lim_{x\to\infty} u(t,x) = 0. \tag{3.18}$$

Sinc interpolations naturally describe the decay at infinity.

REMARK 3.3. Similarly, one can deal with systems of partial differential equations. This generalization, that is also immediate, refers to the case where the dependent variable is a vector. The solution technique leads to a system of equations for each component of the vector variable.

REMARK 3.4. Higher order coefficients can be computed by formulas which exploit recurrence rules. The following result is proposed:

$$a_{ji}^{(2r)} = \frac{(-1)^{i-j}}{h^{2r}(i-j)^{2r}}\sum_{k=0}^{r-1}(-1)^{k+1}\frac{2r!}{(2k+1)!}\pi^{2k}(i-j)^{2k}, \qquad a_{ii}^{(2r)} = \left(\frac{\pi}{h}\right)^{2r}\frac{(-1)^r}{(2r+1)}, \tag{3.19}$$

for even coefficients, where $r = 1,2,\dots$; and

$$a_{ji}^{(2r+1)} = \frac{(-1)^{i-j}}{h^{2r+1}(i-j)^{2r+1}}\sum_{k=0}^{r}(-1)^k\frac{(2r+1)!}{(2k+1)!}\pi^{2k}(i-j)^{2k}, \qquad a_{ii}^{(2r+1)} = 0, \tag{3.20}$$

for odd ones.

4. NONLINEAR BOUNDARY CONDITIONS

The application described in Section 2 shows that some problems in applied sciences need nonlinear boundary conditions. This feature is related to the fact that physical quantities which can be effectively measured at the boundary are not those required by Dirichlet or Neumann statements.

In other words, the statement of linear Dirichlet or Neumann boundary conditions, in applied sciences, may be an idealization of physical reality. Indeed, this is the case of the traffic model which is such that the measurement of the traffic at the boundary involves a nonlinear algebraic expression of the above conditions.

A solution method can be developed by means by simple manipulations of the boundary conditions. First some general techniques are proposed, then the particular case of the traffic flow model is dealt with.

We refer to equation (3.7), where the equations for $i = 2, \dots, n-1$ are still those reported in equation (3.9). The mathematical method has to be developed to obtain the first and last equation from the boundary conditions. Consider then the initial-boundary value problem with boundary conditions stated as follows:

$$\begin{aligned} x &= 0, \qquad \forall\, t \in [0,1] : p(\alpha,\gamma)(t) = a(t), \\ x &= 1, \qquad \forall\, t \in [0,1] : q(\beta,\delta)(t) = b(t), \end{aligned} \tag{4.1}$$

where p, q, a, and b are smooth functions of their arguments. Boundary conditions need to be consistent with the initial condition

$$\begin{aligned} p(\varphi(0), \varphi_x(0)) &= a(0), \\ q(\varphi(1), \varphi_x(1)) &= b(0). \end{aligned} \tag{4.2}$$

Deriving equation (4.1) with respect to time yields

$$\begin{aligned} p_\alpha(\alpha,\gamma)\alpha_t + p_\gamma(\alpha,\gamma)\gamma_t &= a_t, \\ q_\beta(\beta,\delta)\beta_t + q_\delta(\beta,\delta)\delta_t &= b_t. \end{aligned} \tag{4.3}$$

On the other hand, the expressions of γ_t and δ_t can be obtained deriving with respect to time equation (3.13), which yields

$$\begin{aligned} \frac{(-1)^{1-n}}{(1-n)}\,\beta_t + \sum_{j=2}^{n-1} \frac{(-1)^{1-j}}{(1-j)}\, u_{j,t} &= h\,\gamma_t, \\ \frac{(-1)^{n-1}}{(n-1)}\,\alpha_t + \sum_{j=2}^{n-1} \frac{(-1)^{n-j}}{(n-j)}\, u_{j,t} &= h\,\delta_t, \end{aligned} \tag{4.4}$$

where $u_{j,t} = \frac{du_j}{dt}$.

Finally, substituting the expressions of γ_t and δ_t obtained from (4.4) into (4.3) yields

$$\begin{aligned} p_\alpha(\alpha,\gamma)\alpha_t + p_\gamma(\alpha,\gamma)\,\frac{(-1)^{1-n}}{h(1-n)}\,\beta_t &= a_t - p_\gamma(\alpha,\gamma) \sum_{j=2}^{n-1} \frac{(-1)^{1-j}}{h(1-j)}\, u_{j,t}, \\ q_\delta(\beta,\delta)\,\frac{(-1)^{n-1}}{h(n-1)}\,\alpha_t + q_\beta(\beta,\delta)\beta_t &= b_t - q_\delta(\beta,\delta) \sum_{j=2}^{n-1} \frac{(-1)^{n-j}}{h(n-j)}\, u_{j,t}, \end{aligned} \tag{4.5}$$

from which we get, at least formally, α_t and β_t which replace the first and the last equation in the system (3.7).

It can now be shown how the above program works in the case of Problem 2.3, that is with the hydrodynamic model of traffic flow with nonlinear boundary conditions generated by input and output flow measurements. The expressions of p and q are given by the boundary flux measurements.

$$\begin{aligned} p(\alpha,\gamma) &= \left(\alpha - \alpha^2\right)(1 - \eta\gamma), \\ q(\beta,\delta) &= \left(\beta - \beta^2\right)(1 - \eta\delta), \end{aligned} \tag{4.6}$$

and the boundary conditions are defined by the system

$$\begin{aligned} \left[\alpha(t) - \alpha^2(t)\right][1 - \eta\gamma(t)] &= a(t), \\ \left[\beta(t) - \beta^2(t)\right][1 - \eta\delta(t)] &= b(t), \end{aligned} \tag{4.7}$$

where $a(t)$ and $b(t)$ are given smooth functions of time.

Derivation with respect to time yields

$$\begin{aligned} (1-2\alpha)(1-\eta\gamma)\alpha_t - \eta\left(\alpha - \alpha^2\right)\gamma_t &= a_t, \\ (1-2\beta)(1-\eta\delta)\beta_t - \eta\left(\beta - \beta^2\right)\delta_t &= b_t. \end{aligned} \tag{4.8}$$

Substituting the expressions of γ_t and δ_t obtained from (4.4), we have the following system:

$$\begin{aligned} (1-2\alpha)(1-\eta\gamma)\alpha_t - \eta\frac{(-1)^{1-n}}{h(1-n)}\left(\alpha-\alpha^2\right)\beta_t &= a_t + \eta\left(\alpha-\alpha^2\right)\sum_{j=2}^{n-1}\frac{(-1)^{1-j}}{h(1-j)}u_{j,t}, \\ -\eta\frac{(-1)^{n-1}}{h(n-1)}\left(\beta-\beta^2\right)\alpha_t + (1-2\beta)(1-\eta\delta)\beta_t &= b_t + \eta\left(\beta-\beta^2\right)\sum_{j=2}^{n-1}\frac{(-1)^{n-j}}{h(n-j)}u_{j,t}. \end{aligned} \tag{4.9}$$

Solving the above system with respect to α_t and β_t yields

$$\alpha_t = f_1(t,\, u_1, \dots u_n,\, u_{2,t}, \dots u_{n-1,t},\, a_t, b_t) \tag{4.10}$$

and

$$\beta_t = f_n(t,\, u_1, \dots u_n,\, u_{2,t}, \dots u_{n-1,t},\, a_t, b_t), \tag{4.11}$$

where the expressions of f_1 and f_n can be obtained by simple algebraic methods.

Finally, the solution of Problem 2.3 is obtained as the solution of the following system of ordinary differential equations:

$$\frac{du_i}{dt} = f_i(t, u_1, \dots, u_n), \qquad i = 1, \dots, n, \tag{4.12}$$

where

$$f_1 = f_1(t,\, u_1, \dots u_n,\, f_2, \dots f_{n-1},\, a_t, b_t) \tag{4.13}$$

is given by (4.10),

$$\begin{aligned} f_i = (2u_i - 1)&\left[\sum_{j=1, j\neq i}^{n}\frac{(-1)^{i-j}}{h(i-j)}u_j\right] \\ &+ \eta u_i(1-u_i)\left[2\sum_{j=1, j\neq i}^{n}\frac{(-1)^{i-j+1}}{h^2(i-j)^2}u_j - \frac{1}{3}\left(\frac{\pi}{h}\right)^2 u_i\right] \\ &+ \eta(1-2u_i)\left[\sum_{j=1, j\neq i}^{n}\frac{(-1)^{i-j}}{h(i-j)}u_j\right]^2, \end{aligned} \tag{4.14}$$

for $i = 2, \dots, n-1$, and

$$f_n = f_n(t,\, u_1, \dots u_n,\, f_2, \dots f_{n-1},\, a_t, b_t) \tag{4.15}$$

is given by (4.11).

5. PROGRAMMING WITH MATHEMATICA

This section deals with the practical treatment of the mathematical problems described in the preceding sections by generalized Sinc collocation methods and scientific programming by *Mathematica*. The choice of this software is based on several motivations, but mainly on the fact that it is a powerful software which allows the user to operate within an optimal selection of the algorithms without deeply possessing their direct knowledge. The choice of such a software is further discussed and motivated in the last section of this paper.

The reader interested on a deeper information on the potentiality of *Mathematica* is referred to the book by Wolfram [29], as well as to [30] where a large number of problems in classical and continuum mechanics are solved using the above software. Moreover, the reader is addressed to programming with several suggestions and examples.

Here we propose some scientific programs which solve the problems described in the preceding sections. The programs can be downloaded from the following web page:

`http://www.polito.it/~Bellomo.`

Specifically:

- the program *SincCollocation.nb* solves the problem for the traffic flow model given by equation (2.8) with, respectively, Dirichlet boundary conditions (2.10) and nonlinear boundary conditions (2.13);
- the program *DerivNeumannSinc.nb* solves the problem for the traffic flow model given by equation (2.8) with Neumann boundary conditions (2.11).

  The visualization of the results is proposed in Figures 1–6. Specifically we note the following.

  - Figure 1 reports the solution of the Dirichlet problem with initial condition

$$\varphi(x) = 0.7 \times \exp\left[-200 \times (x - 0.3)^2\right] + 0.1 \tag{5.1}$$

  and boundary conditions

$$\begin{aligned} \alpha(t) &= 0.7 \times \exp\left[-200 \times (0.3)^2\right] + 0.1, \\ \beta(t) &= 0.7 \times \exp\left[-200 \times (1 - 0.3)^2\right] + 0.1. \end{aligned} \tag{5.2}$$

  - Figure 2 reports the solution of the Dirichlet problem with initial condition (5.1) and boundary conditions

$$\begin{aligned} \alpha(t) &= 0.7 \times \exp\left[-200 \times (0.3)^2\right] + 0.1 + 0.1 \times t + 0.5 \times t^2, \\ \beta(t) &= 0.7 \times \exp\left[-200 \times (1 - 0.3)^2\right] + 0.1 + 0.2 \times t^2. \end{aligned} \tag{5.3}$$

  - Figure 3 reports the solution of the Neumann problem with initial condition (5.1) and boundary conditions

$$\begin{aligned} \gamma(t) &= -400 \times 0.7 \times \exp\left[-200 \times (0.3)^2\right], \\ \delta(t) &= -400 \times 0.7 \times \exp\left[-200 \times (1 - 0.3)^2\right]. \end{aligned} \tag{5.4}$$

  - Figure 4 reports the solution of the Neumann problem with initial condition

$$\varphi(x) = -2 \times x^2 + 2 \times x + 0.1 \tag{5.5}$$

  and boundary conditions

$$\gamma(t) = 2 \qquad \text{and} \qquad \delta(t) = -2. \tag{5.6}$$

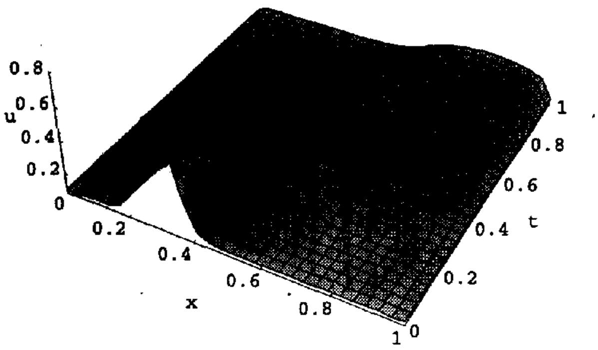

Figure 1.

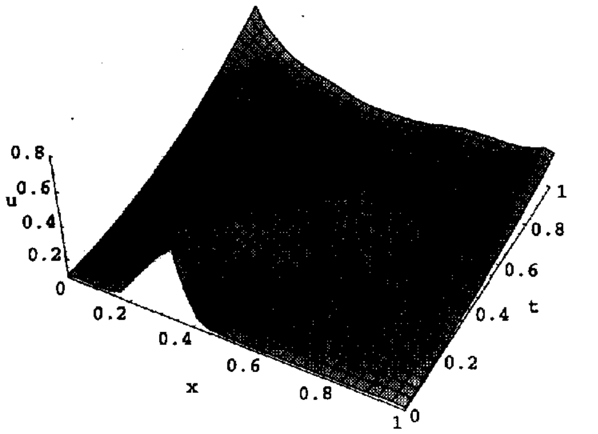

Figure 2.

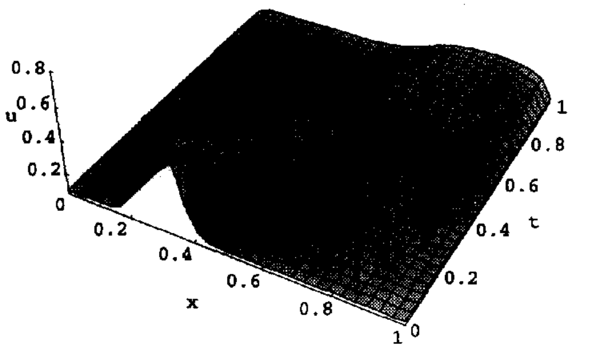

Figure 3.

- Figure 5 reports the solution of the nonlinear boundary conditions problem with initial condition (5.5) and boundary conditions

$$q(t,0) = 0.072 \qquad \text{and} \qquad q(t,1) = 0.108. \tag{5.7}$$

- Figure 6 reports the solution of the nonlinear boundary conditions problem with initial condition (5.1) and boundary conditions

$$q(t,0) = 0.09 + 0.02 \times t^2 \qquad \text{and} \qquad q(t,1) = 0.09. \tag{5.8}$$

These programs are also offered to the reader who wants to solve similar problems related to models different from the ones dealt with in this paper. The necessary technical modifications should be developed without the necessity of a deep knowledge on programming with

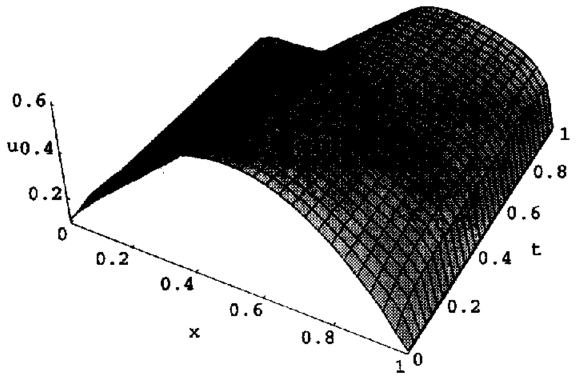

Figure 4.

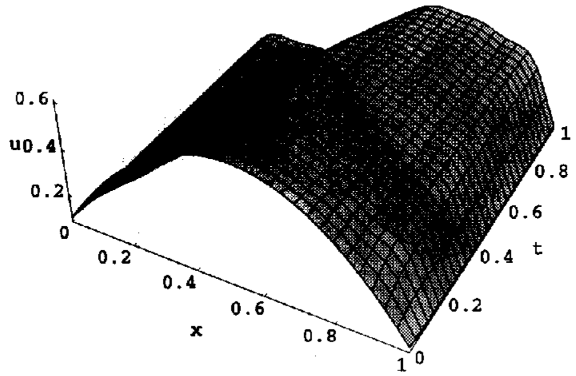

Figure 5.

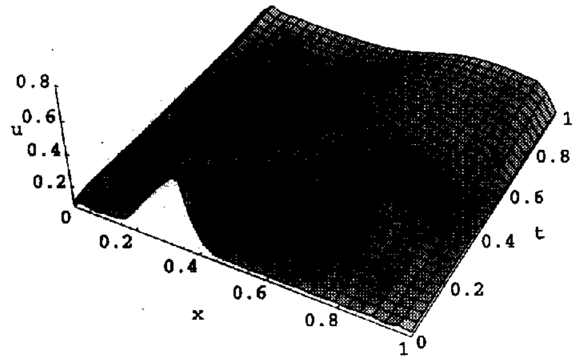

Figure 6.

Mathematica. Certainly, one can also develop alternative codes toward the solution of the same problem. Nevertheless, if we have in mind engineering applications, a specialized software is a good substitute of a great experience in scientific programming.

We remark that the problem of the accuracy of the method is still not discussed yet: it will be in the next section. Moreover, it is useful to develop a critical analysis (as we shall see in the last section) on the mathematical methods also in comparison with other solution techniques. The framework of such critical analysis will allow us to discuss more deeply the utility of scientific softwares.

6. MATHEMATICAL ASPECTS OF SINC INTERPOLATION METHODS

The relevant mathematical problem related to collocation method is the analysis of the approximation errors induced by the application of Sinc interpolation techniques. In particular, one has to define how the above error can be controlled by a proper selection of the collocation points with special attention to spectral approximation. Then one should investigate how the interpolation error propagates in the solution of the problems such as the one we have seen in the preceding paragraphs.

The analysis of the first point can be recovered already in the review paper by Stenger [11]. Additional results are reported in the books [12,13]. The paper by Bonzani [16] organizes the above topic towards the solution of initial boundary value problems.

Considering that Sinc interpolation is defined on the whole real axis, we consider, following [16], a function $v(t,x)$ such that the variable x is defined on the whole real axis $\mathbb{R}$ with v decreasing, for all times, to zero rapidly as $x \to \pm\infty$

$$|v(t;x)| \leq A\, e^{-\alpha|x|}, \qquad \forall\, t \in [0,1], \tag{6.1}$$

where A and α are positive constants. Consider the Sinc interpolation

$$v_N(t;x) = \sum_{i=-N}^{N} v_i(t)\, S_i(x;h), \tag{6.2}$$

with $N > 0$, and S_i defined by (3.3) for $i = -N, \dots, N$. Consider, at fixed time, the following definition of error:

$$e_N = \|E_N(v,h)\|_\infty = \sup_{x\in\mathbb{R}} |E_N(v,h)| = \sup_{x\in\mathbb{R}} |v - v_N|, \tag{6.3}$$

the step h is chosen as follows:

$$h = \frac{k}{\sqrt{N}}, \qquad k > 0. \tag{6.4}$$

As known [11], the error (6.3) is bounded as follows:

$$\|E_N(v,h)\|_\infty \leq C\, e^{-c\sqrt{N}}, \tag{6.5}$$

with C and c positive constants depending on A, α, and k. Analogously the r-derivatives of $v(t,x)$ can be approximated by

$$\frac{\partial^r v}{\partial x^r} \cong v_N^{(r)}(t,x) = \sum_{j=-N}^{N} a_{ji}^{(r)} v_j(t), \tag{6.6}$$

where the interval h is given by equation (6.4). According again to [11], the following estimate holds:

$$e_N^{(r)} = \left\| E_N^{(r)}(v,h) \right\|_\infty = \sup_{x\in\mathbb{R}} \left| v^{(r)} - v_N^{(r)} \right| \leq C_r\, N^{r+(1/2)} e^{-c_r\sqrt{N}}, \tag{6.7}$$

where C_r and c_r are positive constants depending on A, α, k.

It is now immediate transferring the above estimates to functions $v = v(t,x)$, defined over $[0,1] \times \mathbb{R}$, which satisfy inequality (6.1) for every $t \in [0,1]$ and such that

$$v(t,x) = u(t,x), \qquad \forall\, x \in [0,1]. \tag{6.8}$$

If in equation (6.4) the constants N and k (and consequently the step h) are chosen as follows:

$$h = \frac{1}{n+1} = \frac{k}{\sqrt{N}}; \tag{6.9}$$

then for every $t \in [0,1]$, equations (6.3) and (6.5) give the upper bounds for both errors $e_N(v,h)$ and $e_N^{(r)}(v,h)$, related to function v on the whole real axis $\mathbb{R}$, as well as for errors $e_N(u,h)$

and $e_N^{(r)}(u,h)$, referred to the restriction on $[0,1]$ of the same function. In fact, as a consequence of equation (6.8) and of the definition of L_∞-norm, one has

$$\|E_N(v,h)\|_\infty \geq \|E_N(u,h)\|_\infty; \qquad \left\|E_N^{(r)}(v,h)\right\|_\infty \geq \left\|E_N^{(r)}(u,h)\right\|_\infty. \tag{6.10}$$

Then the above approximations have the functional form $O(e^{-c\sqrt{N}})$ for the rate of convergence of the error of an N-points approximation in the space interval $[0,1]$.

The above analysis shows that increasing the number of collocation points decreases monotonically the interpolation error in the desired norm. On the other hand, if one looks at the solution of the initial-boundary value problem, it is immediate to discover that accuracy cannot be obtained increasing indefinitively the number of collocation points. This certainly improves the evaluation of the space derivatives in the collocation points. On the other hand, when the number of equations increases, the system of ordinary differential equations generates further computational errors related to the large dimension of the system.

In general, one should expect that by increasing the number of nodes, the gap between the true solution and the one obtained by the application of the method will gradually decrease up to a certain value n_c. On the other hand, for $n > n_c$, such a gap generally increases. A parallelization method was proposed in [16] in order to overcome the above difficulty.

The sequential steps of application of the method are the following.

Step 1. As first step, the collocation I_x is split into two subcollocations

$$h = 0,\ldots,p+1 : I_1 = \{x_0 = 0,\ldots,x_h,\ldots,x_{p+1} = 1\} \tag{6.11}$$

and

$$k = 0,\ldots,q+1 : I_2 = \{x_0 = 0,\ldots,x_k,\ldots,x_{q+1} = 1\}, \tag{6.12}$$

such that

$$n = p + q, \qquad I_1 \cup I_2 = I_x. \tag{6.13}$$

Step 2. Set the initial value problems corresponding to both collocations in a way that initial conditions are set at $t = t_0$ and boundary conditions are enforced for both systems for $h = k = 0$ and $h = (p+1), k = (q+1)$,

$$\begin{aligned} \frac{du_h}{dt} &= f_h\left(t, x_h, \frac{\partial u}{\partial x}(t,x_h), \frac{\partial^2 u}{\partial x^2}(t,x_h)\right), \\ \mathbf{u}_{h0} &= \varphi(x_h), \end{aligned} \tag{6.14}$$

and

$$\begin{aligned} \frac{du_k}{dt} &= g_k\left(t, x_k, \frac{\partial u}{\partial x}(t,x_k), \frac{\partial^2 u}{\partial x^2}(t,x_k)\right), \\ \mathbf{u}_{k0} &= \varphi(x_k). \end{aligned} \tag{6.15}$$

Step 3. Integrate both systems (6.14) and (6.15) from t_0 to $t_1 = t_0 + \Delta$ using for both of them a third-order Runge-Kutta algorithm

$$\begin{aligned} \mathbf{u}_{h1} &= \mathbf{u}_{h0} + \frac{\Delta}{6}\left(\mathbf{K}_{10} + 4\mathbf{K}_{2h0}(\mathbf{K}_{10}) + \mathbf{K}_{30}(\mathbf{K}_{10}, \mathbf{K}_{2h0})\right), \\ \mathbf{u}_{k1} &= \mathbf{u}_{k0} + \frac{\Delta}{6}\left(\mathbf{K}_{10} + 4\mathbf{K}_{2k0}(\mathbf{K}_{10}) + \mathbf{K}_{30}(\mathbf{K}_{10}, \mathbf{K}_{2k0})\right), \end{aligned} \tag{6.16}$$

where $\mathbf{K}_1$ is computed using all points of the collocation.

Step 4. Once $\mathbf{u}_1$ has been computed, the procedure is repeated starting from Step 2 for all successive integration steps.

The method can be developed also using more than two sub-collocations. Step 1 is modified and consequently in Step 2 one has a number of systems corresponding to the number of collocations. Then the stiffness of the systems of ordinary differential equations is further reduced. Of course the method can be developed also using more accurate integration algorithms, e.g., higher order Runge-Kutta methods. Step 3 is technically modified in order to deal with different algorithms.

The above parallelization algorithm can improve accuracy of the computation, as shown in [16]. However, the computation of the error bounds needs to be carefully studied, not only because it gives a direct information on the level of accuracy of the solution delivered by the mathematical method, but also as it may provide the necessary information to reduce the number of collocation points, and hence the computational time, necessary to obtain the required accuracy.

In general, the error is propagated by the fact that the interpolations are exactly satisfied in the collocation points, but they may be unprecise in the intervals between the collocation points. Moreover, the estimate of the space derivatives is not exact even in the collocation points. Thus, this type of local error generate the gap between the true and the approximated solution delivered by the collocation method.

Before going further on on this matter, we need stating more precisely the concept of error. Consider the space of the functions $u = u(t,x)$ with bounded p-derivative with respect to time for all values of the space variable and bounded q-derivative with respect to space for all values of the time variable. Let $u^n = u^n(t,x)$ the solution delivered by the collocation method and $u = u(t,x)$ the one of the initial-boundary value problem. Suppose that u, u^n belong to a normed space of smooth functions, then the following definition of error can be proposed:

$$e^n = |||u - u^n||| = \sup_{t\in[0,1]} e_\ell(t), \tag{6.17}$$

where

$$e_\ell^n(t) = \|u - u^n\|\,(t) = \max_{x\in[0,1]} |u - u^n|\,(t). \tag{6.18}$$

Actually, a direct estimate of e_ℓ, and hence, of e, can be given only in the case of problems with known analytic solution. Although these problems are often a useful test as documented in [3], problems of interest in applied sciences very unlikely are characterized by analytic solution. In general, one has to start from the error in approximating, at $t = 0$, the space derivatives

$$\varepsilon_n^r = \left\| \frac{\partial^r u}{\partial x^r} - \frac{\partial^r u^n}{\partial x^r} \right\|. \tag{6.19}$$

Then, the estimate of the error propagation can be obtained according to Gronwall's lemma [31].

Several problems addressed to improve accuracy of the solution need to be dealt with. In particular, the following aspects needs to be developed:

- development of best approximation method of the initial datum in order to make as small as possible the initial error ε_n^r,
- development of best interpolation method along the time integration in order to make as small as possible the error e_ℓ^n,
- selection of the number n of collocation points and of the time step of the integration in time,
- improvement of the algorithms for the integration in time,
- development of mathematical methods for the estimates of the error bounds.

The crucial point in the application of the collocation method is the approximation of time dependent functions starting from $t = 0$. Referring to ε_n^r, one has to develop approximations such that

$$\lim_{n\to\infty} \varepsilon_n^r = 0, \tag{6.20}$$

and

$$n > n^*, \quad m > m^* \quad \Rightarrow \quad \varepsilon_m^r < \varepsilon_n^r. \tag{6.21}$$

Polynomial (Lagrange) approximations satisfy the above conditions only if a Chebyscheff collocation is applied. The second condition is often satisfied although it cannot be regarded as a general rule.

Certainly, a good approximation of the initial condition, and subsequently of time dependent conditions, is the starting point to improve accuracy of generalized collocation methods. However, increasing indefinitively the number of collocation points does not imply that e_ℓ^n and e^n will decrease. Indeed, when the number of differential equations become large, integration errors may not be controlled even by reducing the time integration step.

The application of generalized collocation methods need to go through the proper analysis of the following, among several ones, items:

- selection of the type of collocation and of the interpolants;
- selection of the number n of collocation points;
- selection of the time step h of the integration in time.

The first item was already discussed in Section 3. The second and third items, which are somehow related each other, may be quite difficult, or even impossible to be solved theoretically. Therefore, we suggest a computational experimental analysis which may provide useful indications on the above problem. The method is developed through the following steps.

(i) Let n be fixed, then h can be decreased by a step Δh until the distance in norm

$$d_h = |||u_h - u_{h-\Delta h}||| \tag{6.22}$$

tends to a sufficiently small value. The optimum value of h, obtained by the above procedure, corresponding to a certain n is denoted by h_n.

(ii) Consider now the distance in norm

$$d_n = |||u_n - u_{n+1}|||, \qquad h = h_n, \tag{6.23}$$

and increase n until the distance (6.23) tends to a sufficiently small value.

Actually, d_n versus n first decreases (not necessarily with monotonic behaviour) until n reaches a critical value n_c. Then for $n > n_c$ the distance d_n may start to increase. This means that further increasing of n for $n > n_c$ does not give any practical advantage. Thus, considering that the corresponding value of h_n increases more than linearly, one should consider n_c the optimum value of n to be selected for practical calculations.

7. CRITICAL ANALYSIS, GENERALIZATIONS AND DEVELOPMENTS

The contents of this section closes first with a critical analysis and subsequently (also referring to the above critical analysis) with some technical generalizations and developments addressed both to improve the method and to apply it in relatively more general problems. It will be shown that the actual use of *Mathematica* can effectively contribute to solve some of the technical difficulties.

7.1. Critical Analysis

As we have seen, the differential quadrature method can be efficiently applied to the solution of some interesting problems in applied sciences. The method, as already mentioned, was proposed in the paper by Bellman, Casti and Kashef [2,3] and developed and improved by various authors.

Despite the above advantages, also put in evidence in [3], the practical application of the method has to tackle some technical difficulty. Essentially, the main one is that the system of ordinary differential equations obtained by the discretization and interpolation technique are generally stiff due to the fact that each equation contains all variables u_i. This requires the use both of small time steps for the integration with respect to time and sophisticated integration algorithms. The use of *Mathematica* as integration software solves automatically the above problem, so that the monotone decrease of the error bounds, discussed in Section 6, can be effectively exploited.

A question to be naturally posed refers the use of Sinc functions as an alternative to Lagrange polynomials or other types of interpolants. Similarly one should speculate on the technical differences between generalized collocation methods and Galerkin methods. On the other hand, technically minor problems refer to solving problems for systems of equations and problems in several space variables.

The contents of this section are naturally developed according to the above critical analysis and attempt to provide a technical solution to all above problems.

7.2. Lagrange Interpolations and Systems of Equations

The application of the method was described on the basis of the use of Sinc functions. More general, fundamental interpolants, typically Lagrange polynomials, can also be used. The interpolation formula is

$$u(t,x) \cong u^n(t,x) = \sum_{j=1}^{n} L_j(x;n)u_j(t), \tag{7.1}$$

where L_j denote the Lagrange polynomials of order n corresponding to the j^{th} collocation

$$L_j(x;n) = \frac{(x-x_1)\dots(x-x_{j-1})(x-x_{j+1})\dots(x-x_n)}{(x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_n)}. \tag{7.2}$$

Their properties are described in Appendix I of [17] and proposed in [3] for practical applications.

The advantage of the use of Sinc functions rely on the spectral approximation properties described in Section 6. These properties cannot be stated with the same accuracy in the case of Lagrange polynomials. However, in some practical cases, a relatively more satisfactory performance of Lagrange interpolation with respect to the Sinc one can be discovered. A general rule, to be however practically tested, is that one should use Sinc functions for oscillating in space solutions and Lagrange interpolations for nonoscillating solutions. In all cases, the application has to be developed by linking Lagrange type interpolations to the proper Chebyshev collocation. Equispaced collocations are very disappointing although practically proposed in several papers.

Stiffness can be tackled with the splitting algorithms described in Section 6. On the other hand, in some practical cases, but not as a general rule, an efficient solution can be obtained approximating the space derivatives simply using a few nodes on the borders of the collocations rather than all nodes. Generally this requires a relatively larger number of nodes, but practically the method works efficiently, especially with the aid of the optimization of algorithms developed by *Mathematica*.

Dealing with systems of equations is a simple technical problem. If the dependent variable is a vector with N components, one applies, for each component, the Sinc or Lagrange interpolation. The application of the method leads to a system of $N \times n$ ordinary differential equations. Boundary conditions have to be implemented for each component.

7.3. Nonlinear Collocations and Galerkin Methods

As we have seen, generalized collocation methods can be applied using Sinc or Lagrange interpolations. However, it is reasonable to look for improved interpolation techniques. In this line one looks for spectral approximations rather than interpolations. The representation of a function $u = u(t,x)$ is then given by an expansion of the type

$$u(t,x) = \sum_{i=-\infty}^{i=\infty} c_i(t)\psi_i(x), \tag{7.3}$$

where the functions ψ_i belong to a complete set of orthonormal functions defined in a suitable Hilbert space with weighted (the weight function is denoted by $w = w(x)$) inner product

$$\langle f, g\rangle_w(t) = \int f(t,x)g(t,x)w(x)\,dx. \tag{7.4}$$

Consequently, the coefficients c_i are given by $c_i(t) = \langle u, \psi_i\rangle_w$, where the calculations of the integrals may need numerical computation.

If u is *approximated* by u^n corresponding to the collocation I_x, then a truncated expansion is used

$$u(t,x) \cong u^n(t,x) = \sum_{i=-n}^{i=n} c_i(t)\psi_i(x), \tag{7.5}$$

where the coefficients c_i have to be computed numerically exploiting the values of u in the collocation points

$$c_i^n(t) = \sum_{j=-n}^{j=n} W_j u_j^n \psi_i(x_j), \tag{7.6}$$

where W_j denote the weights in the quadrature formula. Replacing the above expression in the the evolution equation yields

$$\begin{aligned}\frac{\partial}{\partial t}\sum_{i=-n}^{i=n} c_i^n(t)\psi_i(x) &= \eta(t,x)\frac{\partial}{\partial x}\sum_{i=-n}^{i=n} c_i^n(t)\psi_i(x) + \mu(t,x)\frac{\partial^2}{\partial x^2}\sum_{i=-n}^{i=n} c_i^n(t)\psi_i(x)\\ &+ \varepsilon f\left(t, x, \sum_{i=-n}^{i=n} c_i^n(t)\psi_i(x), \frac{\partial}{\partial x}\sum_{i=-n}^{i=n} c_i^n(t)\psi_i(x)\right).\end{aligned} \tag{7.7}$$

Taking the inner product yields an equation for the coefficients of the truncated expansion

$$\begin{aligned}\frac{dc_i^n}{dt} &= \sum_{i=-n}^{i=n}\left\langle \eta c_i^n \frac{d\psi_i}{dx}, \psi_i\right\rangle_w + \left\langle \mu c_i^n \frac{d^2\psi_i}{dx^2}, \psi_i\right\rangle_w\\ &+ \varepsilon\left\langle f\left(t, x, \sum_{i=-n}^{i=n} c_i^n\psi_i, \sum_{i=-n}^{i=n} c_i^n \frac{d\psi_i}{dx}\right), \psi_i\right\rangle_w.\end{aligned} \tag{7.8}$$

This procedure may be technically more elaborate than the direct application of collocation methods. However, it may allow selection of more efficient polynomial approximation such as wavelets, see [32,33], and recent developments [34].

7.4. Problems in Two Space Dimensions

Mathematical problems are generally stated in more than one space variable. It is therefore necessary, in dealing with them, to generalize the method which was presented only in the case of one space variable.

The generalization is technical. However, one has to tackle the difficulties involved by the large number of equations. Initial boundary value problems in two space dimensions can be solved by techniques analogous to the ones used for the problems described in Section 3. In order to avoid repetitions, we will simply indicate how the interpolation can be organized, then a few guidelines will be described in order to show the solution method exploiting the collocation interpolation described in Section 2. Consider equations in two space dimensions, that may be written as follows:

$$\frac{\partial u}{\partial t} = f\left(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 u}{\partial x\partial y}, \dots\right), \tag{7.9}$$

where u is the dependent variable

$$u = u(t,x,y) : [0,1] \times [0,1] \times [0,\ell] \to [-1,1], \tag{7.10}$$

or, more in general

$$u = u(t,x,y) : [0,1] \times D \to \mathbb{R}, \tag{7.11}$$

where the boundary of D will be denoted by ∂D.

As in one-dimensional cases, we consider the following problems.

PROBLEM 7.1. Consider the initial-boundary value problem for equation (7.9) with initial condition

$$u(0,x,y) = \varphi(x,y), \qquad \forall x,y \in D, \tag{7.12}$$

and Dirichlet boundary conditions

$$\forall t \in [0,1], \quad \forall x,y \in \partial D : u(t;x,y) = \alpha^*(t), \quad x,y \in \partial D, \tag{7.13}$$

given as smooth functions consistent, for $t = 0$, with the initial condition (7.12).

PROBLEM 7.2. Consider the initial-boundary value problem for equation (7.9) with initial condition (7.12) and Neumann boundary conditions

$$\forall t \in [0,1], \quad \forall x,y \in \partial D : \mathbf{n} \cdot \nabla u(t;x,y) = \gamma^*(t), \quad x,y \in \partial D, \tag{7.14}$$

given as smooth functions of time, where $\mathbf{n}$ is the normal to the boundary ∂D.

Considering that the solution schemes are simply technical developments of those already seen in Section 3, their presentation will be very concise. The same interpolation can be used for time dependent functions in two space variables. Consider first the case such that the independent variables are defined over the rectangle $[0,1] \times [0,\ell]$. This means that $u = u(t,x,y)$ is defined over $[0,1] \times [0,1] \times [0,\ell]$. We also assume that $u(t;x,y)$ is a one-to-one map from $[0,1] \times [0,\ell]$ into $[0,1]$, for every $t \in [0,1]$.

Bearing this in mind, consider, in addition to the collocation I_x, the following one:

$$j = 1,\dots,m : I_y = \{y_1 = 0,\dots,y_j,\dots,y_m = \ell\}. \tag{7.15}$$

It follows that the dependent variable can be interpolated and approximated as follows:

$$u = u(t,x,y) \cong u^{nm}(t,x,y) = \sum_{i=1}^{n}\sum_{j=1}^{m} S_i(x;h)S_j(y;h)u_{ij}(t). \tag{7.16}$$

The approximation of the space derivatives in the collocation points is obtained by calculations analogous to the ones of the one-dimensional case:

$$\frac{\partial^r u}{\partial x^r}(t;x_i,y_j) \cong \sum_{h=1}^{n} a_{hi}^{(r)} u_{hj}(t), \qquad \frac{\partial^r u}{\partial y^r}(t;x_i,y_j) \cong \sum_{k=1}^{m} a_{kj}^{(r)} u_{ik}(t), \tag{7.17}$$

while mixed type derivatives are given by formula of the type

$$\frac{\partial^2 u}{\partial x \partial y}(t;x_i,y_j) \cong \sum_{h=0}^{n+1}\sum_{k=0}^{m+1} a_{hi}^{(1)} a_{kj}^{(1)} u_{hk}(t), \tag{7.18}$$

where the coefficients a are given by the expressions reported in Section 3 and depend on the number of collocation points.

The interpolation yields, following the same procedure we have seen in Section 3, to a system of $n \times m$ ordinary differential equations corresponding to the nodal point of the collocations (x_i, y_j).

The solution of the initial-boundary value problems is obtained linking to such equation the initial conditions are $\varphi(x_i, y_j)$ and boundary conditions. These are imposed implementing the given time dependent values of the dependent variable

$$u_{ij}(t) = \alpha^*(t), \qquad x_i, y_j \in \delta D, \tag{7.19}$$

in the case of Problem 7.1 and the system obtained by equation (7.14) in the case of Problem 7.2.

In principle, both interpolations, Sinc or Lagrange, can be used. The technical alternatives are the following:

$$u = u(t,x,y) \cong u^{nm}(t,x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} L_i(x;n) L_j(y;m) u_{ij}(t), \tag{7.20}$$

$$u = u(t,x,y) \cong u^{nm}(t,x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} S_i(x;h) L_j(y;m) u_{ij}(t), \tag{7.21}$$

and

$$u = u(t,x,y) \cong u^{nm}(t,x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} L_i(x;n) S_j(y;h) u_{ij}(t). \tag{7.22}$$

The method can be developed in a semi-infinite strip: $x, y \in \mathbb{R}_+ \times [0, \ell]$ or $x, y \in [0,1] \times \mathbb{R}_+$; or, still in two dimensions if both variables are defined in the half-space $x, y \in \mathbb{R}_+ \times \mathbb{R}_+$. The method can be developed applying Remark 3.2 in two space dimensions.

One has to be careful in dealing with problems such that the dependent variables are defined on nonrectangular domains. In fact, to a point of the discretization of one variable may correspond more than one point of the other variable. In addition, the main problem still consists in dealing with a large number of stiff differential equations. Therefore, the various methods to reduce stiffness, described in Section 6 and Section 7.1, have to be taken into account.

REFERENCES

1. R. Bellman and J. Casti, Differential quadrature and long term integration, *J. Math. Anal. Appl.* **34**, 235–238, (1971).
2. R. Bellman, B. Kashef and J. Casti, Differential quadrature: Solution of nonlinear partial differential equations, *J. Comp. Phys.* **10**, 40–52, (1972).
3. C. Bert and M. Malik, Differential quadrature method in computational mechanics, *ASME Review* **49**, 1–27, (1996).
4. N. Bellomo, Nonlinear models and problems in applied sciences: From differential quadrature to generalized collocation methods, *Mathl. Comput. Modelling* **26** (4), 13–34, (1997).
5. A. Satofuka, A new explicit method for the solution of parabolic differential equations, In *Numerical Methodologies and Properties in Heat Transfer*, pp. 97–108, Hemisphere, New York, (1983).
6. N. Bellomo and F. Flandoli, Stochastic partial differential equations in continuum physics, *Mathl. Comp. in Simul.* **31**, 3–17, (1988).
7. C. Bert, S. Jang and A. Stritz, Nonlinear bending analysis for orthotropic rectangular plates by the method of differential quadrature, *Comp. Mech.* **5**, 217–226, (1988).
8. L. Preziosi and L. de Socio, A nonlinear inverse phase transition problem for the heat equation, *Modelling Math. Methods in Appl. Sci.* 1, 167–182, (1991).
9. I. Bonzani, Time evolution of random fields in stochastic continuum mechanics, *Mathl. Comput. Modelling* **17** (6), 37–45, (1993).
10. C. Bert, S. Jang and A. Stritz, Static and free vibrational analysis of beam and plates by the method of differential quadrature, *Acta. Mech.* **102**, 12–24, (1994).
11. F. Stenger, Numerical methods based on Whittaker cardinal or Sinc functions, *SIAM Review* **23**, 165–224, (1981).
12. J. Lund and K. Bowers, *Sinc Methods*, SIAM Publ., Philadelphia, (1992).
13. F. Stenger, *Numerical Methods Based on Sinc and Analytic Functions*, Springer, Berlin, (1993).

14. N. Bellomo and L. Ridolfi, Solution of nonlinear initial-boundary value problems by Sinc collocation-interpolation methods, *Computers Math. Applic.* **29** (4), 15–28, (1995).
15. E. Longo, G. Teppati and N. Bellomo, Discretization of nonlinear models by Sinc colloca- tion-interpolation methods, *Computers Math. Applic.* **32**, 68–11, (1996).
16. I. Bonzani, Solution of nonlinear evolution problems by parallelized collocation interpolation methods, *Computers Math. Applic.* **34** (12), 71–79, (1997).
17. N. Bellomo and L. Preziosi, *Modelling, Mathematical Methods and Scientific Computation*, CRC Press, Boca Raton, FL, (1995).
18. A.A. Samarskii *et al.*, Invariant difference schemes for parabolic equations with transformation of the independent variables, *Math. Models Methods in Appl. Sci.* **9**, 93–110, (1999).
19. R. Sandboge, Adaptative finite element methods for reactive compressible flow, *Math. Models Methods in Appl. Sci.* **9**, 211–242, (1999).
20. S. Adjerid, J. Flaherty and I. Babuska, *A posteriori* error estimates for the finite element method of lines solution of parabolic equations, *Math. Models Methods in Appl. Sci.* **9**, 261–286, (1999).
21. J.P. Vila, On particles weighted methods and smooth particle hydrodynamics, *Math. Models Meth. Appl. Sci.* **9**, 161–210, (1999).
22. B. Di Martino, F.J. Chatelon and P. Orenga, The nonlinear Galerkin method applied to shallow water equations, *Math. Models Meth. Appl. Sci* **9**, 825–854, (1999).
23. S.A. Lifits, S.Yu. Reutskiy and B. Tirozzi, Quasi Trefftz spectral method for Stokes problem, *Math. Models Meth. Appl. Sci* **7**, 1187–1212, (1997).
24. N. Bellomo, L. Preziosi and A. Romano, *Mechanics and Dynamic Systems with Mathematica*, Birkhäuser, Boston, (2000).
25. W. Leutzback, *Introduction to the Theory of Traffic Flow*, Springer, Berlin, (1996).
26. A. Klar, Mathematical models for vehicular traffic, In *Generalized Boltzmann Models in Applied Sciences*, (Edited by N. Bellomo and M. Pulvirenti), Birkhäuser, Boston, (2000).
27. D. Helbing, A. Hennecke, V. Shvetsov and M. Treiber, Micro- and macro-simulation of freeway traffic, *Math. Comput. Modelling* (to appear).
28. E. De Angelis, Nonlinear hydrodynamic models of traffic flow modelling and mathematical problems, *Mathl. Comput. Modelling* **29** (7), 83–95, (1999).
29. S. Wolfram, *Mathematica: A System for Doing Mathematica by Computer*, Addison Wesley, (1992).
30. A. Marasco and A. Romano, *Ordinary Differential Equations with Mathematica*, Birkhäuser, Boston, (2001).
31. E. Zeider, *Applied Functional Analysis*, Springer, Berlin, (1995).
32. I. Daubechies, *Ten Lectures for Wavelets*, SIAM Publication, Philadelphia, (1992).
33. Y. Meyer, *Wavelets Algorithms and Applications*, SIAM Publication, Philadelphia, (1993).
34. S.G. Talocia and A. Tabacco, Wavelets on the interval with optimal localization, *Modelling Math. Methods in Appl. Sci* **10**, 441–461, (2000).
35. E. Zeider, *Applied Functional Analysis*, Springer, Berlin, (1995).